数理解析研究所講究録
906 巻 1995 年 220-227

正データからの Simple Regular 言語の多項式時間帰納推論

佐藤清朗　　佐藤優子
大阪府立大学 総合科学研究科

1995 年 2 月 2 日 (1994 年度 冬の LA シンポジウム)
於 京都大学数理解析研究所

Abstract

本稿では, simple regular 言語と呼ばれる正則言語の部分族が正データから多項式時間帰納推論可能であることを示す. この族は Tanida and Yokomori の導入した very regular 言語族を真に含むことが示される.

1 はじめに

帰納推論とは, 与えられたデータからそれを説明する一般的な規則を推測する過程である. Gold [5] は「極限における同定」という帰納推論の成功基準を導入し, 形式言語や帰納的関数の帰納推論に関してその理論的基盤を築いた. 今日, 帰納推論の研究は計算論的学習理論の一研究分野として活発な研究がなされている.

本稿では, 極限同定の枠組みで正データからの正則言語の帰納推論について論じる. 正データからの帰納推論に関して, Gold [5] は超有限な言語族 (すべての有限言語と少なくとも 1 つの無限言語を含む言語族) が帰納推論可能ではないことを示した. これは, Chomsky の階層の最下層に位置し, 最も基本的な言語である正則言語の族でさえも正データからは帰納推論可能ではないということを意味する. しかし, Angluin [3] は有限証拠集合という概念を用いて正データからの推論可能性の特徴付け定理を与え, パターン言語等の興味深い言語が正データから帰納推論可能であることを示した. 本稿で扱う正則言語に関しては, word 上の正則表現で演算回数を高々 n に制限した正則言語族の部分族が正データから帰納推論可能であることが示されている (Sato [8]).

一方, 現実的な問題を考える場合, 推論可能であるだけでなく「効率的な推論」ということを考慮に入れなければならない. 極限同定では, 推論過程が無限に続くため, 推論機械が推測を更新する時間を効率の尺度とみなすのが一般的である. そこで, データが入力されてから推測を出力するまでの計算時間がそれまでの入力の長さに関する多項式時間であるような推論機械が存在するとき, 多項式時間推論可能であるといい, 効率よく推論することと同一視する. 最近では, 正則言語の多項式時間推論可能性が最も基本的でかつ重要な問題として多くの研究がなされてきており, reversible 言語 [4] や, very regular 言語 [9] などが正データから多項式時間推論可能であることが示されている.

本稿では, simple regular と呼ばれる word 上の決定性オートマトンを導入し, それで受理される言語が正データから多項式時間推論可能であることを示す. この言語族は, Tanida & Yokomori [9] による very regular 言語の族を真に包含し, Angluin [4] による reversible 言語族とは, incomparable である.

2 準備

この節では, 形式言語の正データからの帰納推論に関する基本的な定義と, これまでに得られている結果についてまとめる.

Σ をアルファベット (alphabet) と呼ばれる空でない有限集合とする. Σ 上の word を $w, u, v, u_1, \cdots$ などで表し, Σ 上の言語 (language) を $L, L_1, L_2, \cdots$ などで表す. $N = \{n \mid n \geq 1\}$ と置く.

定義 2.1 言語族 $\mathcal{L} = L_1, L_2, \cdots$ が帰納的言語の添字付き族 (indexed family of recursive languages) であるとは, 次のような計算可能関数 $f : N \times \Sigma^* \to \{0,1\}$ が存在することを言う:

$$f(i,w) = \begin{cases} 1, \ w \in L_i \text{ のとき} \\ 0, \ w \notin L_i \text{ のとき} \end{cases}$$

以下, 言語族は帰納的言語の添字付き族と仮定する.

定義 2.2 word の無限列 $w_1, w_2, \cdots$ が言語 L の正提示 (positive presentation) であるとは, $L = \{w_i \mid i \in N\}$ であることを言う.

推論機械とは, ときどき入力を要求し, ときどき出力 (推測) を生成するような実行的手続きである. 推論機械 M が 正データから L を帰納推論する (infer L from positive data) とは, L の任意の正提示を M に与えたとき, M の生成する推測の列が $L = L_j$ となる j に収束することを言う. このとき, 推論機械 M は言語 L を極限において同定する (identify L in the limit) と言う.

定義 2.3 言語族 $\mathcal{L}$ が正データから帰納推論可能 (inductively inferable) であるとは, 任意の言語 $L \in \mathcal{L}$ を正データから帰納推論する推論機械 M が存在することである.

どのような言語族が正データから帰納推論可能となるのであろうか? Gold [5] は超有限な言語族 (すべての有限言語と少なくとも 1 つの無限言語を含む言語族) が帰納推論可能ではないことを示した. これは, Chomsky の階層の最下層に位置する正則言語の族でさえも (超有限であるために) 正データからは帰納推論可能ではないことを意味する. しかし, Angluin [3] は次に述べる有限証拠集合という概念を導入し, 言語族が正データから帰納推論可能であることを特徴付ける定理を与えた.

定義 2.4 $T \subseteq \Sigma^*$ 及び, $L \in \mathcal{L}$ とする. 集合 T が (言語族 $\mathcal{L}$ での) 有限証拠集合 (finite tell-tale, 以下 ftt) であるとは, (i) T が L の有限部分集合でかつ, (ii) $T \subseteq L' \subsetneq L$ となる言語 $L' \in \mathcal{L}$ が存在しないことをいう.

定理 2.1 [3] 言語族 $\mathcal{L} = L_1, L_2 \cdots$ が正データから帰納推論可能であるための必要十分条件は, 任意の i に対し言語 L_i の ftt を枚挙する実行的な (effective) 手続きが存在することである.

この定理は, 言語の ftt の存在が正データから帰納推論可能であるための必要条件であることを意味する. パターン言語族 [1][3] や, 有限言語の族 [5] など, 正データから推論可能な興味深い言語族が多々示されている. 本稿は, 正則言語の正データからの帰納推論を扱う. 前に述べたように正則言語の族は正データから帰納推論可能でない. ではどのような正則言語族の部分族が推論可能であろうか. Sato [8] は, 正則言語族の部分族である. Σ^* の word に $\cup, \cdot, *$ の演算を高々 n 回施して得られる言語の族 ($\mathcal{L}_\Sigma(n)$ と書く) が正データから帰納推論可能であることを示した. 明らかに $\bigcup_{n=0}^{\infty} \mathcal{L}_\Sigma(n)$ は正則言語族と一致する.

定理 2.2 [8] 言語族 $\mathcal{L}_\Sigma(n)$ は正データから帰納推論可能である.

帰納推論の実際的な場面では推論可能が保証されただけでは不十分である. 効率的な推論時間が重要な鍵となる. 本稿では次に述べる多項式時間極限同定の問題を扱う.

定義 2.5 言語族 $\mathcal{L}$ が正データから多項式時間帰納推論可能であるとは, 次の条件を満たす推論機械 M が存在することである.
(i) M は $\mathcal{L}$ の任意の言語を正データから帰納推論する.
(ii) M は i 番目のデータを受取り, i 番目の結果を出力し, その計算時間がそれまでの入力の長さに関する多項式時間である.

正則言語に関してこれまでに zero-reversible 言語族 [4], szilard 言語族 [6], very regular 言語族 [9] などが, 多項式時間推論可能であることが知られている.

3 SR 言語

head(v) で word v の先頭の文字を表すものとする.

定義 3.1 $W \subseteq \Sigma^+$が prefix-free であるとは, W の任意の 2 つの word が互いに prefix にならないことを言う. 特に, W のすべての word の head が互いに異なっているとき, W は simple prefix-free (以下, SPF) 集合と言う.

以下, W は simple prefix-free 集合 とする.

定義 3.2 word 上の決定性有限オートマトン $M = (Q, W, \delta, p_0, F)$ が simple regular オートマトン (SRA) であるとは M が次の条件を満たすことである:
(1) Q は状態の有限集合. (2) W は simple prefix-free 集合. (3) $p_0 \in Q$ は初期状態. (4) $F \subseteq Q$ は受理状態の集合. (5) δ は $Q \times W$ から Q への部分関数で, 次の条件を満たす:

任意の $u \in W$ に対し $\delta(p, u) = q$ となる $p, q \in Q$ が存在し, q はただ 1 つ定まる.

δ の定義域は $Q \times W$ から $Q \times W^*$ へ拡張されているものとする. $p \in Q$ に対して, 遷移 $\delta(p, u) = q$ $(q \in Q,\ u \in W)$ が存在するとき, これを p からの遷移という. q への遷移或いは u による遷移も同様に定義する. SRA では条件 (5) より, 任意の $u \in W$ に対して $\delta(p, u) = q$ を満たす $q \in Q$ は唯一である. この q を u に対応する状態といい q_u と書く.
SRA M により受理される言語 $L(M)$ を simple regular (SR) 言語と呼び, すべての SR 言語の族を $\mathcal{L}(\mathrm{SR})$ で表す.

定義 3.3 $M = (Q, W, \delta, p_0, F)$ が k-very regular オートマトン (k-VRA) であるとは次の条件をみたすことである:
(1) W は SPF でかつ, $W \subseteq \Sigma^{\leq k}$.
(2) 任意の $u \in W$ に対して, $\delta(p, u) = q$ を満たすペア (p, q) がただ一つ存在する.
(3) 任意の $p, q \in Q$ に対して, $\delta(p, u) = q$ を満たす $u \in W$ は高々一つしか存在しない.

k-VRA が受理する言語を k-very regular 言語と呼び, その族を $\mathcal{L}_k(\mathrm{VR})$ と書く.
$\mathcal{L}(\mathrm{VR}) = \bigcup_{k=1}^{\infty} \mathcal{L}_k(\mathrm{VR})$ と置き, 言語 $L \in \mathcal{L}(\mathrm{VR})$ を very regular 言語と呼ぶ.

定理 3.1 $\mathcal{L}(\mathrm{VR}) \subsetneq \mathcal{L}(\mathrm{SR})$

proof. $L \in \mathcal{L}(\mathrm{VR})$ とする. $L(M) = L$ となる VRA $M = (Q, W, \delta, p_0, F)$ が存在する. 次の方法で, $L(M) = L(M')$ を満たす SRA $M' = (Q', W, \delta', p_0, F')$ を構成する : $W = \{u_1, u_2, \cdots, u_m\}$ と置く.
(i)$Q' = \{p_0, p_{u_1}, p_{u_2}, \cdots, p_{u_m}\}$.
(ii)δ' : $i = 1, 2, \cdots, m$ に対して $\delta(p_0, u_i) = q$ となる $q \in Q$ が存在するとき, $\delta'(p_0, u_i) = q_{u_i}$. また, 任意の $i, j (1 \leq i, j \leq n)$ に対して, $\delta(p, u_i u_j) = q$ となる $p, q \in Q$ が存在するとき, $\delta'(p_{u_i}, u_j) = q_{u_j}$ とする.

$$
\text{(iii)}\ F' = \begin{cases} \{p_{u_i} \in Q' \mid \exists p \in Q,\ \exists q \in F,\ \exists i \ni \delta(p, u_i) = q\}, & p_0 \notin F. \\ \{p_{u_i} \in Q' \mid \exists p \in Q,\ \exists q \in F,\ \exists i \ni \delta(p, u_i) = q\} \cup \{p_0\}, & p_0 \in F. \end{cases}
$$

M' の構成の仕方から, M' は明らかに SRA である. 以下, $L(M) = L(M')$ を示す.

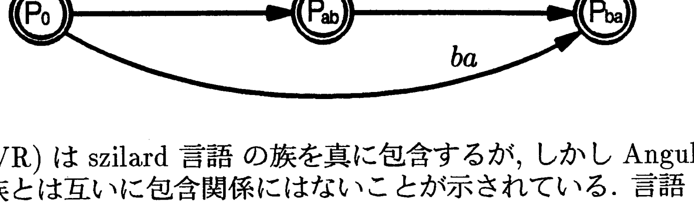
$w = v_1 v_2 \cdots v_l \in L(M)$, $(v_i \in W, i = 1, 2, \cdots, l)$.

$\iff$ $\delta(p_{j-1}, v_j) = p_j$ $(j = 1, 2, \cdots, l)$ となる遷移が存在し, $p_l \in F$.

$\iff$ $\delta'(p_{v_{j-1}}, v_j) = p_{v_j}$ $(j = 1, 2, \cdots, l)$ となる遷移が存在し, $p_{v_l} \in F'$, ただし, $p_{v_0} = p_0$.

$\iff$ $w \in L(M')$.

従って, $\mathcal{L}(\mathrm{VR}) \subseteq \mathcal{L}(\mathrm{SR})$ が成り立つ.
次に, 言語 $L = \{(ab)^m (ba)^n \mid m, n \geq 0\}$ を考える, L は 次の SRA で受理されるので SR 言語である. しかし very regular 言語ではない (cf.[9]). この例より $\mathcal{L}(\mathrm{VR}) \subsetneq \mathcal{L}(\mathrm{SR})$ が成り立つ.



論文 [9] で, $\mathcal{L}(\mathrm{VR})$ は szilard 言語 の族を真に包含するが, しかし Anguluin [4] が定義した zero-reversible 言語の族とは互いに包含関係にはないことが示されている. 言語 $L = \{a^m b a^n \mid m, n \geq 0\}$

は zero-reversible 言語であるが SR 言語でない. 故に族 $\mathcal{L}(\mathrm{SR})$ も zero-reversible 言語の族とは互いに包含関係にはない. 定理 3.1から $\mathcal{L}(\mathrm{SR})$ も szilard language の族を真に包含する.

4 SR 言語の正データからの帰納推論

4.1 Reduced SRA

$M=(Q,W,\delta,p_0,F)$ を SRA とする. 状態列 $p_1,p_2,\cdots,p_l$ が word $w\in W^+$ に対応する (Mでの) 列というのは, $\delta(p_i,u_i)=p_{i+1},(i=1,2,\cdots,l-1)$ であることを言う. ただし, $w=u_1u_2\cdots u_l$ で $u_i\in W$ $(i=1,2,\cdots,l)$. 状態 $p\in Q$ に対して, $w\in W^+$ が 状態 p での simple-loop-word であるとは, w に対応する状態列 $p_1,p_2,\cdots,p_l$ が存在し, $p_i\neq p_j$ $(i\neq j,\ 1\leq i,j\leq l-1)$ でかつ, $p=p_1=p_l$ であることを言う. wが状態 p で loop-free であるとは, w に対応する状態列 $p_1,p_2,\cdots,p_l$ が存在し, $p_i\neq p_j$ $(i\neq j,\ 1\leq i,j\leq l)$ であることを言う. 状態 $p\in Q$ に対し Pre, Post, Loop を次のような word の集合と定義する.
(1) $\mathrm{Pre}(p)=\{w\in W^+\mid \delta(p_0,w)=p,\ w$は$p_0$で loop-free$\}$.
(2) $\mathrm{Post}(p,p_f)=\{w\in W^+\mid \delta(p,w)=p_f,\ w$は$p$ で loop-free$\}$, $p_f\in F$.
(3) $\mathrm{Post}(p)=\cup_{p_f\in F}\mathrm{Post}(p,p_f)$.
(4) $\mathrm{Loop}(p)=\{\lambda\}\cup\{w\in W^+\mid w$は$p$ で simple-loop-word$\}$.
状態 $p\in Q$ に対して, $\mathrm{outdegree}(p)=\#\{u\in W\mid \exists q\in Q \ni \delta(p,u)=q\}$.
$\mathrm{indegree}(p)=\#\{u\in W\mid \exists q\in Q \ni \delta(q,u)=p\}$.

定義 4.1 SRA $M=(Q,W,\delta,p_0,F)$ が reduced であるとは次の三つの条件を満たすことである:
(1) $\mathrm{indegree}(p_0)=0$.
(2) $F\cup\{p_0\}$ を除く任意の状態 $p\in Q$ に対し, $\mathrm{outdegree}(p)>1$.
(3) 任意の $p\in Q$ に対して, $\mathrm{Pre}(p)\mathrm{Post}(p)\neq\emptyset$.

補題 4.1 任意の SR 言語 L に対し, $L(M)=L$ を満たす reduced SRA M が存在する.

proof. $M=(Q,W,\delta,p_0,F)$ を $L(M)=L$ となる SRA とする. 次の手続きで $L(M')=L(M)$ となる reduced SRA $M'=(Q',W',\delta',p_0,F')$ を構成する:
先ず, $\mathrm{Pre}(p)\mathrm{Post}(p)=\emptyset$ を満たす状態 $p\in Q$ が存在するならば, p を Q から取り除き, さらに p を含む遷移を全て取り除く. また p に対応する word を W から取り除く. そうして得られる SRA を改めて M とする. 明らかに受理される言語は変わらない.

次に, 初期状態 p_0 への遷移 $\delta(p,u)=p_0$ $(p\in Q,\ u\in W)$ が存在する場合. このような u は唯一である. p_u を M で使用されていない状態とし, u による p_0 への各遷移をすべて u による p_u への遷移に置き換え, p_0 からの各遷移$\delta(p_0,v)=q$ $(q\in Q,\ v\in W)$ に対して, $\delta(p_u,v)=q$ を付け加える. もし, $p_0\in F$ なら p_u も F につけ加える.

最後に, $\mathrm{outdegree}(p)=1$ $(p\in Q-F,\ p\neq p_0)$ となる p が存在する場合. p からの 唯一の遷移を $\delta(p,v)=q$ $(v\in W,\ q\in Q)$ とし, p を $u\in W$ に対応する状態とする. この時 uv を一つの word とみなし W 中の u を uv と置き換え, $\delta(p,v)=q$ を取り除く. そして$\delta(p',u)=p$ となる遷移をすべて $\delta(p',uv)=p$ で置き換える. $\delta(q,v')=q'$ $(q'\in Q,\ v'\in W)$ となる遷移が存在するなら, $\delta(p,v')=q'$ をつけ加える. (v による q への遷移が一つだけの場合は, v を W から取り除き, q も Q から取り除く. さらに q からの遷移もすべて取り除く.) 以上のことを, $\mathrm{outdegree}(p)=1$ $(p\in Q-F,\ p\neq p_0)$ となる $p\in Q$ が存在しなくなるまで繰り返す.

以上の手続きで構成された SRA M' は明らかに $L(M')=L(M)$ を満たす. ■

4.2 SR 言語の特徴付け集合

定義 4.2 $\mathcal{L}$ を言語族, $L\in\mathcal{L}$ とする. $S\subseteq\Sigma^*$ が ($\mathcal{L}$ での)L の特徴付け集合 (characteristic sample) とは, S が L の有限部分集合でかつ, L が S を含む $\mathcal{L}$ での最小言語であることをいう.
S が L の特徴付け集合ならば, S は L の ftt である. しかし, 逆は一般には成り立たない. また, S が L の特徴付け集合ならば, 明らかに $S\subseteq S'\subseteq L$ となる有限集合 S' は L の特徴付け集合である.

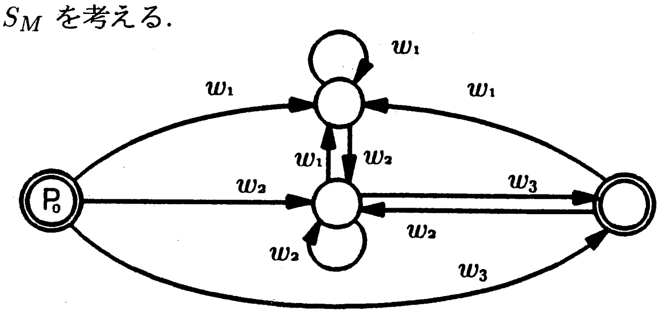
以下, SRA M に対する $L(M)$ の特徴付け集合について述べる.

$M=(Q,W,\delta,p_0,F)$ を SRA とする. $L(M)$ の有限部分集合 S_M を次のように定義する.

$$S_M=\cup_{p\in Q}\mathrm{Pre}(p)\mathrm{Loop}(p)\mathrm{Post}(p).$$

S_M は $L(M)$ の有限部分集合で, 部分語として simple-loop-word を高々一つ含む word の集合となる.

例 4.1 SRA に対する S_M を考える.



各状態の Pre(p), Loop(p), Post(p) は次の通りである.
(1) $p_0; \{\lambda\}\{\lambda\}\{\lambda, w_1w_2w_3, w_2w_3, w_3\}$.
(2) $p_1; \{w_1\}\{\lambda, w_1, w_2w_1, w_2w_3w_1\}\{w_2w_3\}$.
(3) $p_2; \{w_2, w_1w_2\}\{\lambda, w_2, w_1w_2, w_3w_1w_2, w_3w_2\}\{w_3\}$.
(4) $p_3; \{w_3, w_1w_2w_3, w_2w_3\}\{\lambda, w_2w_3, w_1w_2w_3\}\{\lambda\}$.

従って, S_M は次の集合で与えられる.

$$S_M = \{w_1w_2w_3, w_2w_3, w_3, w_1w_1w_2w_3, w_1w_2w_1w_2w_3, w_1w_2w_3w_1w_2w_3, w_2w_2w_3, w_2w_1w_2w_3, w_2w_3w_1w_2w_3, w_2w_3w_2w_3, w_1w_2w_2w_3, w_1w_2w_3w_2w_3, w_3w_2w_3, w_3w_1w_2w_3\}$$

S_M が $L(M)$ の特徴付け集合であることを示す前に, SPF 集合 W に関する結果を述べる.

補題 4.2 W を SPF 集合とする. $wv_1, wv_2 \in W^*$ かつ head$(v_1) \neq$ head(v_2) ならば, $w, v_1, v_2 \in W^*$ である. ただし $w, v_1, v_2 \in \Sigma^*$ とする.

定義 4.3 SPF 集合 W が言語 L の生成集合とは, $L \subseteq W^*$ かつ任意の $W' \subsetneq W$ に対して $L \not\subseteq (W')^*$ となることを言う. L の生成集合の族を $\mathcal{W}^L$ と書く.

2 つの SPF 集合 W と W' に対して次のような順序関係を導入する.

定義 4.4 W と W' を SPF 集合とする.
$W \preceq W' \iff W \subseteq (W')^+$. 明らかに関係 $\preceq$ は半順序関係である.

補題 4.3 [11] S が Σ^* の有限部分集合ならば, $(\mathcal{W}^S, \preceq)$ は束集合である.

補題 4.4 $M = (Q, W, \delta, p_0, F)$ が reduced SRA ならば $W = W_{inf}^{S_M}$ が成り立つ. ただし, $W_{inf}^{S_M} = \inf \mathcal{W}^{S_M}$.

proof. $u \in W$, u に対応する状態を p_u とする. $\delta(q, u) = p_u$ となる $q \in Q$ $(q \neq p_u)$ が存在する. M は reduced であるから Pre$(q) \neq \emptyset$. w を Pre(q) の word とする.

(I) $wu \in (W_{inf}^{S_M})^+$ であることを示す.
$p_u \notin F$の時. outdegree$(p_u) > 1$ より head$(w_1) \neq$ head(w_2) となる $w_1, w_2 \in$ Post(p_u) が存在する. 明らかに $wuw_i \in S_M (i = 1, 2)$ である. 従って, $wuw_i \in (W_{inf}^{S_M})^+ (i = 1, 2)$. head$(w_1) \neq$ head(w_2) であるから補題4.2 より $wu \in (W_{inf}^{S_M})^+$.
$p_u \in F$ の時は, $wu \in S_M$ より, 明らかに $wu \in (W_{inf}^{S_M})^+$.

(II) $u \in (W_{inf}^{S_M})^+$ であることをを示す.
$q = p_0$ ならば. $w = \lambda$ であるから (I) より $u \in (W_{inf}^{S_M})^+$.
$q \in F$ ならば $w \in S_M$. 従って $w \in (W_{inf}^{S_M})^+$ より $u \in (W_{inf}^{S_M})^+$.
最後に, $q \notin F (q \neq p_0)$ とする. outdegree$(q) > 1$ より q からの遷移 $\delta(q, u') = p_{u'}$ $(u' \in W,\ u \neq u',\ p_{u'} \in Q)$ が存在する. (I) と同様にして $wu' \in (W_{inf}^{S_M})^+$ が示される.
head$(u) \neq$ head(u') より補題4.2 に用いると $w, u, u' \in (W_{inf}^{S_M})^+$.

(I),(II) より $u \in (W_{inf}^{S_M})^+$. すなわち $W \preceq W_{inf}^{S_M}$. ■

補題 4.5 $M=(Q,W,\delta,p_0,F)$ と $M'=(Q',W',\delta',p'_0,F')$ を reduced SRA とする．
$S_M\subseteq L(M')$ ならば, $W\preceq W'$.

proof. $S_M\subseteq L(M')$ より, $S_M\subseteq W'^*$. 従って $W'\in\mathcal{W}^{S_M}$. 一方, 補題 4.4より $W^{S_M}_{inf}=W\preceq W'$.■

補題 4.6 $M=(Q,W,\delta,p_0,F)$ と $M'=(Q',W',\delta',p'_0,F')$ を reduced SRA とする．
$S_M\subseteq L(M')$ ならば, 次のような条件を満たす関数 $f:Q\longrightarrow Q'$ が存在する:

$$\delta(p,u)=q(p,q\in Q,\ u\in W)\text{ ならば }\delta'(f(p),u)=f(q).$$

proof. $Q=\{p_0\}\cup\{p_u\mid u\in W\}$ と置く．Q から Q' への関数 f を次のように定義する:
(1) $f(p_0)=p'_0$ とする．
(2) 任意の $u\in W$ に対して, 補題 4.5 より $W\preceq W'$ であるから, $u\in W'^+$. 従って $u=v_1v_2\cdots v_l$ $(l\geq 1)$ となる $v_1,v_2,\cdots,v_l\in W'$ が存在する．M' は reduced SRA であるから, v_l に対応する状態 $q'_{v_l}\in Q'$ は一意である．この q'_{v_l} に対して $f(p_u)=q'_{v_l}$ とする．
この定義から明らかに $f(F)\subseteq F'$ が成り立つ．
　以下, この関数 f は補題の条件を満たすことを示す．
$\delta(p,u)=p_u$ $(p,p_u\in Q,\ u\in W)$ とする．
(I) $p=p_0$ の場合．$w'\in\mathrm{Post}(p_u)$ とすると $uw'\in S_M$. 仮定より $uw'\in L(M')$. $u\in W'^+$ であるから, M' に u に対応する状態列が存在し, 最後の状態は q'_{v_l}. $f(p_0)=p'_0$ であるので $\delta'(f(p_0),u)=f(p_u)$.
(II) $p=p_v$ $(v\in W)$ の場合．$w\in\mathrm{Pre}(p_v)$, $w'\in\mathrm{Post}(p_u)$ とすると $wuw'\in S_M$. 仮定より $wuw'\in L(M')$. 一方, $w=w_1v$ $(w_1\in W^*)$ と表されるので, M' に w_1vu に対応する状態列が存在し, $\delta'(p'_0,w_1vu)=\delta'(f(p_v),u)=f(p_u)$ が成り立つ．■

補題 4.7 $M=(Q,W,\delta,p_0,F)$ を reduced SRA とする．この時 S_M は, $L(M)$ の特徴付け集合である．

proof. $M'=(Q',W',\delta',p'_0,F')$ を $S_M\subseteq L(M')$ を満たす reduced SRA とする．以下, $L(M)\subseteq L(M')$ を示す．$w=u_1u_2\cdots u_l\in L(M)$ $(u_i\in W,\ i=1,2,\cdots,l)$ とする．M での遷移 $\delta(p_{u_{i-1}},u_i)=p_{u_i}$ $(i=1,2,\cdots n,\ p_{u_0}=p_0)$, $p_{u_n}\in F$ が存在する．$S_M\subseteq L(M')$ より, 補題 4.6 の条件を満たす関数 $f:Q\longrightarrow Q'$ が存在する．従って M' で $\delta'(f(p_{u_{i-1}}))=f(p_{u_i})$ $(i=1,2,\cdots n,\ f(p_{u_0})=f(p_0))$, かつ $f(p_{u_n})\in F$ である．$f(p_0)=p'_0$ より $\delta'(p'_0,w)=f(p_{u_n})\in F'$. 故に, $w\in L(M')$.■
　任意の reduced SRA M に対し, S_M は計算可能であるから, 定理 2.1 及び, 補題 4.7より次の定理が成り立つ．

定理 4.1 $\mathcal{L}(\mathrm{SR})$ は正データから帰納推論可能である．

4.3 SR 言語の多項式時間極限同定

　この節では, 有限個の正のサンプルからまず W を構成し, 次に W 上の reduced SRA を構成する方法を与え, 最終的な推論 algorithm を求める．
以下, $\Sigma=\{a_1,a_2,\cdots,a_m\}$, L を reduced SR 言語及び, $S=\{w_1,w_2,\cdots w_n\}$ を L の有限部分集合とする．
まず有限集合 S から S を生成する SPF 集合を求める方法を述べる．
任意の SPF 集合 W は m 個の成分を持つベクトル $\mathcal{T}=(t_{a_1},t_{a_2},\cdots,t_{a_m})$ で表される．ただし, $t_{a_j}(1\leq j\leq m)$ は λ もしくは a_j で始まる W の word である．次の手続き (Tanida and Yokomori [9] を引用) は, 新しい例 $x\in\Sigma^*$ が入力された時, x を生成できるように現在の $\mathcal{T}$ を更新するプログラムである．
　common-prefix(t_{a_j},x); t_{a_j} と x の最長の共通 prefix word.
　suffix(t_{a_j},x); $t_{a_j}=xy$ を満たす suffix y.

```
procedure UPDATE(T, x)
begin
    if x ≠ λ then begin
        Let a_j := copy(x, 1, 1);            (文字列 x の先頭の文字を a_i とする)
        if t_{a_j} = λ then t_{a_j} := x;
        else begin
```

```
            z := common-prefix(t_{a_j}, x);
            u := suffix(z, x);
            v := suffix(z, t_{a_j});
            t_{a_j} := z;
            call UPDATE(T, u);
            call UPDATE(T, v);
        end
    end
end
```

以下, S の $\mathcal{T}$ により分割された λ を除く各成分の集合を W_S で表すことにする.

補題 4.8 [11] $\mathcal{T}$ は入力の順序によらない.

次に, S と W_S から次のように reduced SRA $M_S = (Q, W_S, \delta, p_0, F)$ を構成する: 遷移 $\delta(p, w) = q$ を単に (p, w, q) と表し, δ を (p, w, q) の集合として扱う.

```
procedure CONSTRUCT(T, S)
begin
    Q := {p_0};
    repeat
        input w ∈ S;
        q := p_0;
        if w = λ then stock p_0 in F;
        else begin
            devide w = u_1 u_2 ··· u_l, (1 ≤ i ≤ l, u_i ∈ W_S);      (入力 wを W_Sにより分解する)
            for i= 1 to l
                begin
                    stock (q, u_i, p_{u_i}) in δ , stock p_{u_i} in Q;
                    q := p_{u_i}, stock q in Q;                          (手前の状態を記憶)
                end
            stock p_{u_l} in F;
        end
    untill S = {∅}
end
```

M_S の構成法及び補題 4.8より次の結果が成り立つ.

補題 4.9 L を SR 言語とする. S,T が $S \subseteq T \subseteq L$ を満たす有限集合ならば, $L(M_S) \subseteq L(M_T) \subseteq L$.

補題 4.10 L を SR 言語とする. S が L の特徴付け集合ならば, $L(M_S) = L$.

次に正のサンプルから SR 言語を推論する Algorithm を示す. この Algorithm は (Tanida and Yokomori[9] を引用) 入力として SR 言語 L の正提示を与え, それに対応する SRA を出力する.

```
Identification Algorithm IA
begin
    S := ∅;
    T := (λ, ···, λ);
    M_S := (∅, ∅, ∅, p_0, ∅);
    repeat
        let M_S = (Q_S, W_S, δ_S, p_0, F_S) be the current SRA;
        read the next positive example w;
        S := S ∪ {w};
        if w ∈ L(M_S) then output M_S ;
        else begin
            call UPDATE(T, w);
```

call CONSTRUCT($\mathcal{T}, S$);
output M_S ;
end
untill $S = \{\emptyset\}$
end

補題 4.11 任意の SR 言語 L と L の任意の正提示 $w_1, w_2, \cdots$ に対して, Algorithm IA の出力の列 $M_{S_1}, M_{S_2}, \cdots, M_{S_i}, \cdots$ は次を満たす. ただし, $S_i = \{w_1, w_2, \cdots w_i\}$ $(i = 1, 2, \cdots)$.
(1) $\forall i \geq 1, L(M_{S_i}) \subseteq L(M_{S_{i+1}}) \subseteq L$
(2) $\exists n_0 \geq 1 \ni \forall r \geq n_0,\ M_{S_r} = M_{S_{n_0}} \wedge L(M_{S_r}) = L$

proof. 任意の i に対して, 補題 4.9より $L(M_{S_i}) \subseteq L(M_{S_{i+1}}) \subseteq L$. 一方, 定理 4.2 より L を受理する reduced SRA M が存在する. S_M は L の有限部分集合であるから, $S_M \subseteq S_{n_0}$ を満たす $n_0 \geq 1$ が存在する. 定理 4.7より, S_M は L の特徴付け集合である. 従って, 補題 4.10より, $L(M_{S_M}) = L(M_{S_{n_0}}) = L$. 故に, 任意の $r \geq n_0$ に対して $M_{S_r} = M_{S_{n_0}}$.■

補題 4.10及び, 補題 4.11から次の定理を得る.

定理 4.2 Algorithm IA は, 任意の SR 言語 L に対して, $L = L(M)$ を満たす reduced SRA M を正データから極限同定する.

$S = \{w_1, w_2, \cdots, w_{i-1}\}$ とし, IA により生成された出力を $M_S = (Q_S, W_S, \delta_S, p_0, F_S)$ とする. 新しい例 w_i が与えられて、推測 M_S を更新する時間を考察する.

Tanida and Yokomori[9] で示されているように, UPDATE($\mathcal{T}, w_i$) の complexity は $O(l(m+1))$ である. ただし, $l = \max_{1 \leq j \leq i} |w_j|$.

$S' = S \cup \{w_i\}$ に対して, CONSTRUCT($\mathcal{T}, S'$) の complexity は i 個の入力例のそれぞれについて, $W_{S'}$ で分解した数に依存するので, $K = \sum_{1 \leq j \leq i} |w_j|$ とすると, $O(K)$ である. したがって, Algotithm IA における time complexity は $O(Km)$ である.

定理 4.3 Algorithm IA は, 任意の SR 言語に対して, $L = L(M)$ を満たす reduced SRA M を正データから多項式時間極限同定する.

参考文献

[1] Angluin, D., "*Finding patterns common to a set of strings*", Proc. 11th Annual Symposium on Theory of Computing, pp. 130–141, 1979.

[2] Angluin, D., "*On the complexity of minimum inference of regular sets*", Information and Control, vol. 39, pp. 337–350, 1978.

[3] Angluin, D., "*Inductive inference of formal languages from positive data*", Information and Control, vol. 45, pp. 117–135, 1980.

[4] Angluin, D., "*Inference of reversible language*", Journal of the Association for Computing Machinary, vol. 29, No. 3, pp. 741–765, July 1982.

[5] Gold, E. M., "*Language idetification in the limit*", Information and Control, vol. 10, pp. 447–474, 1967.

[6] Mäkinen, E., "*The grammatical inference problem for the szilard languages of linear grammars*", Information Processing Letters, vol. 36, pp. 203–206, 1990.

[7] Pitt, L., "*Inductive inference, DFAs, and computational complexty*", Proc. Workshop on Analogical and Inductive Inference(Lecture Notes in Computer Science, 397), pp. 18–44, Springer-Verlag 1989.

[8] Sato, M., "*Inductive inference of formal languages*", Bull. Inf. Cybern., 1995 to appear.

[9] Tanida, N. and Yokomori, T., "*Polynomial-time identification of very regular languages in the limit*", Proc. 2nd Workshop on Algorithmic Learning Theory, pp. 61–72, 1991.

[10] Yokomori, T. and Ishida, N., "*Learning local languages and its application to protein α-chain prediction*", Proc. 27th Hawaii International Conference on System Science, vol. V, pp. 113–122, 1994.

[11] 渡辺 紀仁・佐藤 優子, "完全データからの多項式時間反駁推論可能な Simple Regular 言語族について", February(1995), 京大数理解析研究所講究録「アルゴリズムと計算量理論」.